

From graphs to tables

Yaron Koren

SMWCon Fall 2016

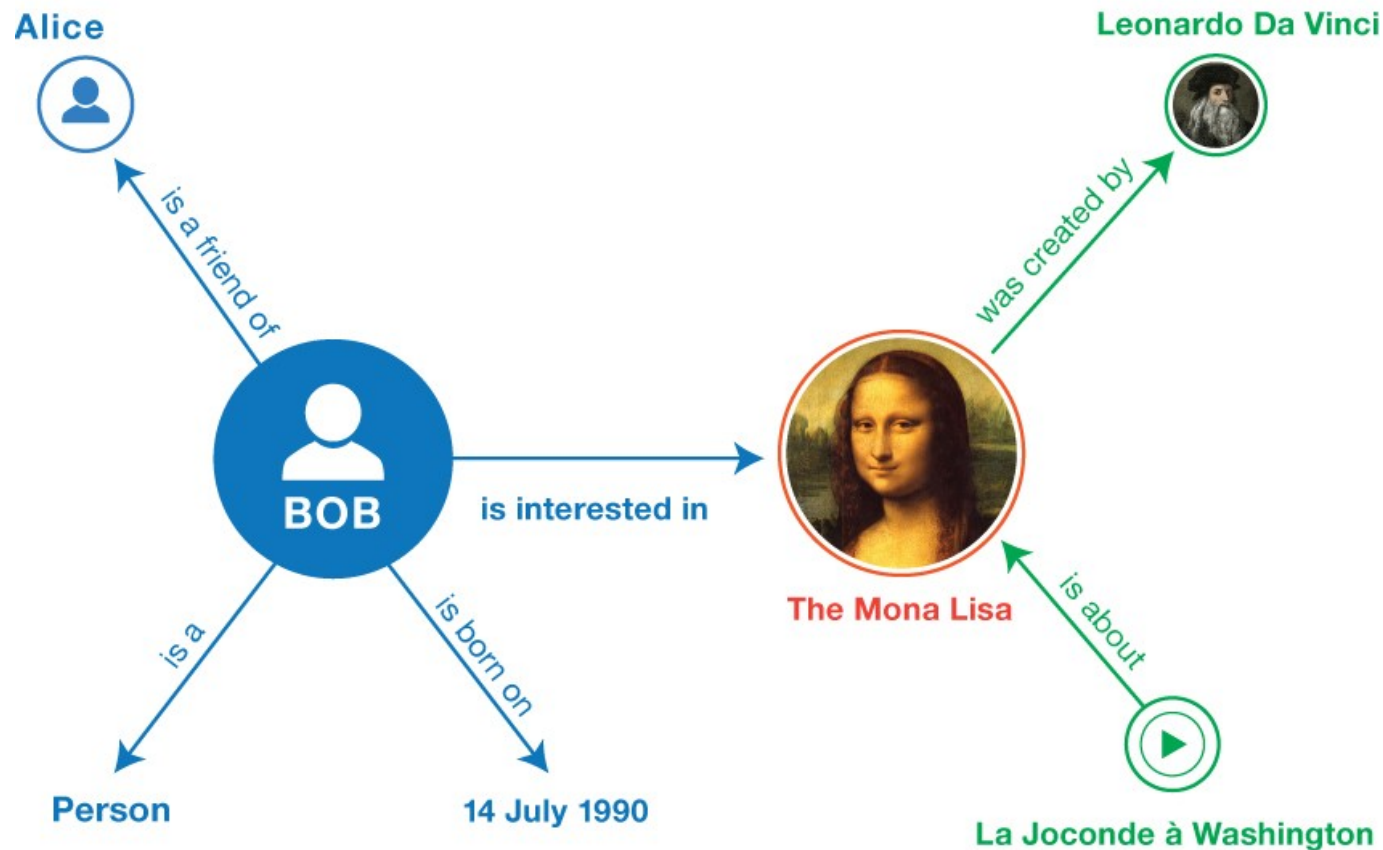
September 29, 2016

Frankfurt, Germany

About me

- Live in New York City
- “Enterprise MediaWiki” developer, consultant, hoster and author
- My consulting company: WikiWorks (<http://wikiworks.com>)
- My book: *Working with MediaWiki* (<http://workingwithmediawiki.com>)

Sample graph: from the W3C “RDF Primer”



Sample table

Tourists to Australia (2006)		
Country of origin of visitors	Number of visitors	Average length of stay (nights)
Italy	51 737	42
China	308 452	48
United States of America	456 084	24
United Kingdom	734 244	34
Canada	109 843	42
New Zealand	1 075 797	14

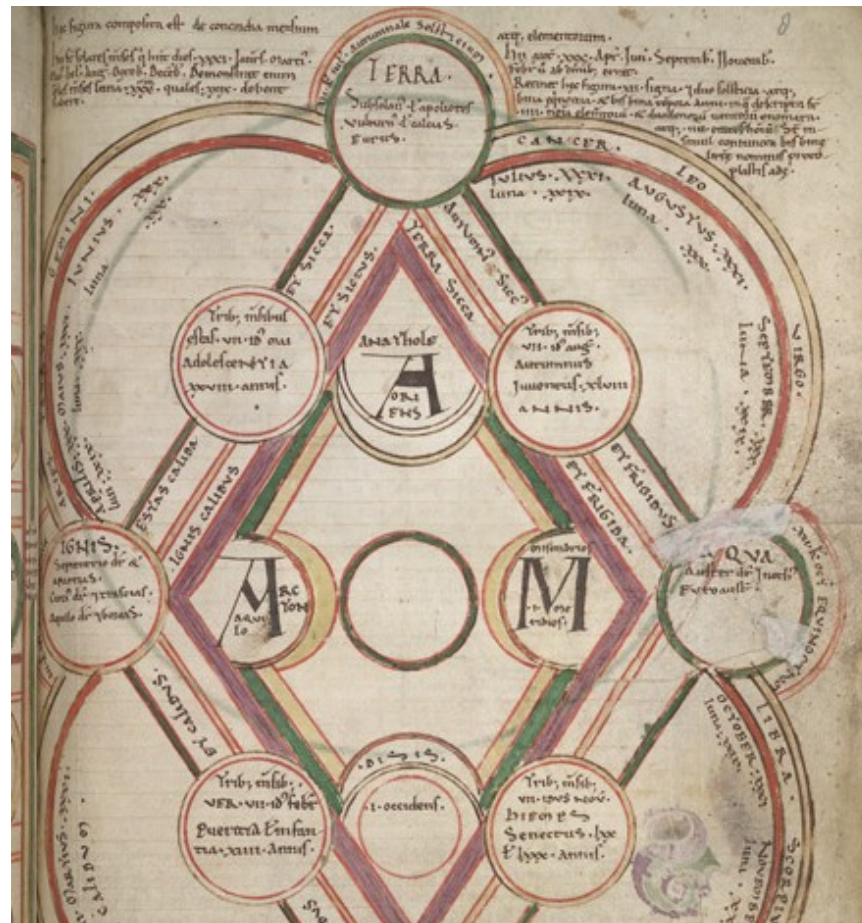
Any data set can be stored as either a
graph or a table.

Both tables and graphs have a **long history.**

From the Temple of Karnak in Egypt, ~ 1500 BC



By the British monk Byrhtferth, ~ 1000 AD



Storage options

Tables:

- Relational databases
- Spreadsheets
- CSV
- ...and: calls to infobox templates in MediaWiki

Graphs:

- RDF triplestores
- RDF/XML

MW templates define tables

- Every call to a template is a **row**
- Every template parameter is a **column**
- Every individual value is a **table cell**

(Let's ignore templates that can take arbitrary parameters, using Scribunto/Lua...)

RDBMS/SQL vs. RDF/SPARQL

Relational DBs	RDF/SPARQL
Better for computer-generated data	Better for real-world data
Enforcement of structure	No structure; all data allowed
Since 1970s	Since 2000s
Available on vast majority of web servers	Requires custom installation

The Semantic MediaWiki approach

A split approach:

- Usually holds infobox template data, i.e. *tables*
- Data is represented and queried as *triples*
- Usually stored in a relational DB (*tables*)

Because the SMW backend is (usually) not a triplestore and not quite a relational DB, you can't use either SQL or SPARQL on it.

Instead: the #ask query language.

Limitations of #ask

- Can't query on null/missing values
- Can't do AND + OR
- ~~Can't query linked properties (“? Property A.Property B”)~~ (*coming in SMW 2.5*)
- No string operations (substring, concat, length)
- ...etc.

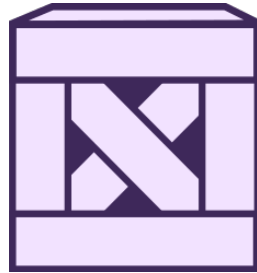
All of these *are* doable with both SQL and SPARQL.

This is not surprising.

Creating/maintaining a custom query language is a lot of work.

...and #ask has to translate to both SQL and SPARQL.

Cargo



A MediaWiki extension I created in 2015.

<https://www.mediawiki.org/wiki/Extension:Cargo>

An alternative to Semantic MediaWiki.

The Cargo approach

Data is stored in “true tables”.

Each **template** is stored in **its own DB table** – tables are re-generated after a schema change.

SQL-like querying.

“SQL endpoint”

Cargo may have originated the “SQL endpoint” - a URL where you can (more or less) call SQL SELECT queries directly.

(Special:ViewData)

Another SPARQL-like approach.

Side note...

Cargo now can match almost all of the SMW, etc. basic functionality.

Recent addition: the “EditNotify” extension.

EditNotify

A MediaWiki extension created by Abhinand N. as part of the 2016 Google Summer of Code

<https://www.mediawiki.org/wiki/Extension:EditNotify>

Lets you get emails when a specified template field is changed, etc.

A “non-semantic” equivalent of Semantic Watchlist

Cargo and RDF/SPARQL

Still missing in Cargo: RDF output,
SPARQL querying.

This could be added – Cargo can be
made “Semantic”.

Back to graphs & tables...

A Cargo-like store for Wikidata data would be a useful *addition* (not replacement) for the triplestore.

Does not have to be an official Wikimedia project.

Advantages of a Relational DB store for Wikidata

- 1) Enables a drill-down interface (?)
- 2) SQL-based querying will let many more people run their own queries
- 3) May lead to greater structuring within Wikidata

About drill-down...

RDF triplestores can't create “temporary graphs”.

Semantic Drilldown (and Cargo's equivalent, Special:Drilldown) make heavy use of temporary tables.

Is a high-speed drill-down interface
impossible with an RDF/SPARQL triplestore?

I don't know, but... maybe.

This may become an issue on Wikipedia,
if/when Wikidata data replaces categories.

(Historical note: replacing categories on
Wikipedia was one of the main original
goals of Semantic MediaWiki)

From <https://en.wikipedia.org/wiki/Category:Films>:

- ▶ [Films by city](#) (1 C)
- ▶ [Films by country](#) (226 C)
- ▶ [Films by language](#) (207 C)
- ▶ [Films by audience](#) (3 C)
- ▶ [Films by continent](#) (6 C)
- ▶ [Films by culture](#) (16 C)
- ▶ [Films by date](#) (6 C)
- ▶ [Films by director](#) (7 C)

This makes sense for replacement by a faceted drill-down interface.

Some unsolicited advice for
the Wikidata developers

Wikidata should store a **mapping** between **classes** (i.e., “instance of” values) and **properties** for that class.

Can be done by either defining “domains” for properties, or allowed properties for classes.

“Domains” in Wikidata

This info is already stored in property talk pages.

Example – for “capital”

(https://www.wikidata.org/wiki/Property_talk:P36):

Domain	administrative territorial entity (Q56061)
Allowed values	human settlement (Q486972), mainly capital (Q5119) (note:

However, this is unofficial and unused.

Why set domains for properties?

- Allows for translating data into tables
- Allows for defining drill-down filters (unrelated to actual implementation)
- Prevents editing errors

Wikidata could benefit from having many of its property “constraints”, currently listed in talk pages, be stored officially.

Examples: **data type, domain, allowed values, allowed units**

What about exceptions?

Erika Eiffel (born Erika Labrie):

<https://www.wikidata.org/wiki/Q509934>

spouse



Eiffel Tower

► 1 reference

This is not an ideal example...

- Relates to the *range* of a property, not its *domain*
- Maybe this value should just be removed (sounds like a non-consensual relationship)

Handling exceptions to data constraints in Wikidata

My suggestion: show a warning icon/tooltip.

That way, edge cases will be allowed, but editors (and readers) will know that something is out of the ordinary.



Overall message

Flexible structure > rigid structure

Flexible structure > no structure

Credits

Thanks to Markus Krötzsch and Denny Vrandečić
for their input on this talk.

(Does not imply an endorsement.)

Questions/comments/complaints